DENSE INTEREST FEATURES FOR VIDEO PROCESSING

Roeland De Geest and Tinne Tuytelaars

KU Leuven ESAT-PSI, iMinds
{roeland.degeest, tinne.tuytelaars}@esat.kuleuven.be

ABSTRACT

We propose two novel feature detection methods for action recognition, based on the dense interest points described by Tuytelaars [1]. The first one is an extension of dense interest points to three dimensions. In the second one, trajectories are constructed starting from dense interest points. We present an analysis of the properties of these methods and conclude that both give higher classification accuracies than dense sampling when less features are used.

***Index Terms*—** Action classification, Video representation

1. INTRODUCTION

The first step in many image and video analysis systems is feature detection. Especially in the context of object or action classification, feature selection and representation is a crucial step. In image processing, the first feature detectors were designed to focus on certain characteristic image points like corners (e.g., Harris corner detector [2]). Later, it was shown that better image classification could be obtained by sampling regularly on a dense grid in the image [3]. This way, the whole image is covered and less information is lost.

In video processing, feature detection evolved similarly. Laptev and Lindeberg [4] extended the Harris corner detector to include the temporal dimension. This way, *spatio-temporal interest points* (STIP) are detected: image corners undergoing a change in their motion. Other interest point detectors were derived similarly from Gabor filters [5] or the Hessian matrix [6]. Dense sampling was later introduced by Wang et al. [7] and proved to outperform interest points on video processing as well.

Dollár et al. [5] stress that the temporal dimension has different characteristics than the spatial dimensions. Therefore, it seems logical to treat it differently, and not simply extend the 2D detectors to 3D. Interest point tracking is a logical choice: interest points are well defined, and tracking succeeds in capturing their motion. Different methods have been proposed to capture and describe those trajectories for action recognition, e.g., [8, 9]. Later, Wang et al. [10, 11] introduced *dense trajectories*: they no longer start the trajectories from detected interest points, but from a densely sampled grid instead. They prove that, for action recognition, dense trajectories outperform interest point detectors as well as other trajectory based methods.

Although dense sampling gives better classification results because it covers the whole image or video, interest point based methods have some advantages as well. An interest point is easier to locate: two different images or videos from an object or action will likely have their interest points on similar locations. Tuytelaars [1] developed a hybrid system to detect features in 2D images, that combines elements of interest points and dense sampling. Her dense interest point detector outperforms both approaches. In this paper, we investigate whether a similar method can be applied to videos.

We briefly explain 2D dense interest points in section 2. In section 3 and 4 we introduce two novel video feature detection methods, the first one based on a 3D interest point detector and the second one based on trajectories. We evaluate them in section 5 and conclude in section 6.

2. BACKGROUND: DENSE INTEREST POINTS

Our work is based on the 2D dense interest points developed by Tuytelaars [1]. She designed a hybrid scheme to combine the advantages of dense sampling with those of interest points. Her method starts from a dense grid over multiple octaves and multiple scales within these octaves. The grid points are refined however to let them align somewhat with the interest points of the image. A maximum over the output of an interest point detector function (e.g., the Laplacian of Gaussian) is searched in the neighbourhood of each point, spatially as well as over scales. The point neighbourhoods touch each other to allow a large deviation from the original grid positions. The found maxima are referred to as *dense interest points* (DIPs): they are better localized than the grid points, but still cover the whole image.

These DIPs are shown to outperform dense sampling and interest points on Pascal VOC2007 [12]. In the remainder of this paper, we extend them to video processing.

This work was supported by the Flemish Fund for Scientific Research (FWO) through the project *Multi-camera human behavior monitoring and unusual event detection* and by the FP7 ERC Starting Grant 240530 *COGNIMUND*.

3. SPATIO-TEMPORAL DENSE INTEREST POINTS

In our first approach, we consider the video as a 3D image. We extend Tuytelaars's DIPs to the spatio-temporal domain, similar to Laptev's extension of the Harris corner detector [4]. First, we apply a Gaussian filter on the video in each of the three dimensions. Afterwards, we use a 3D Laplacian to obtain the LoG of the videos.

A 2D LoG detects blobs. In 3D, the most significant local maxima in the LoG response occur at the place a blob passes by. Other local maxima are found on spatial non-moving blobs or when an object with non-blob shape is moving.

To enable scale selection, we filter the original video with multiple Gaussian filters, each with a different sigma. In our setup, we use two values for sigma in space and two values for sigma in time, thus obtaining four different responses for a video. More values lead to a better scale selection, but the calculation time increases considerably.

Finally, the filtered video is divided in a grid of 4D cuboids. Each cuboid has width and height equal to w, depth (expressed in number of frames) equal to t and as fourth dimension the number of scales s within an octave. With dense sampling the centers of the cuboids would be selected as features. On the other hand, our *spatio-temporal dense interest points* (STDIPs) correspond to the maximum filter response inside this cuboid. We take $s = 4$. It is however also possible to take s smaller than the number of calculated scales. This would increase the number of cuboids (and as a consequence the number of STDIPs), but the precision of the scale selection decreases unless the LoG response is calculated for more values of sigma (which increases the calculation time).

In our experiments we repeat this detection process over three octaves. With these parameters we use as many scales as [4]. We calculate a 72-dimensional HOG and a 90-dimensional HOF descriptor with Laptev's code[1] [7] so we can compare our STDIPs directly with his STIPs.

An example of different spatio-temporal features can be found in Figure 1. STIPs are only found on the moving person, while the dense points regularly cover the whole image. STDIPs however are centered on the person and the edges in the image. Both STIP and STDIP are capable of optimizing the feature positions in time.

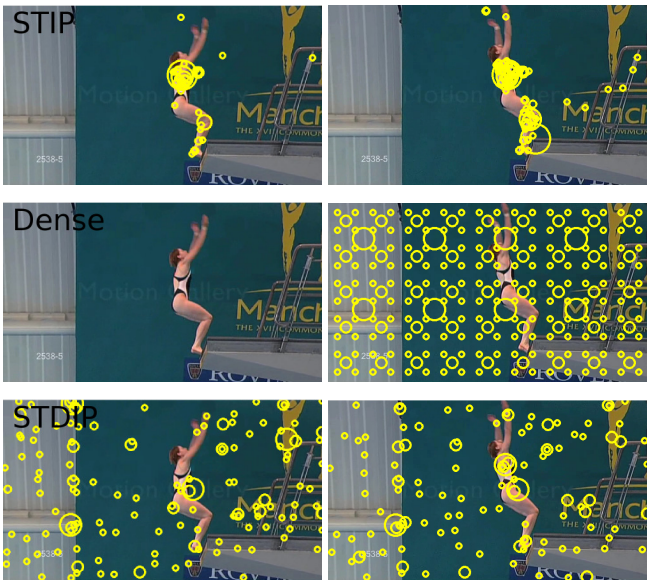

Fig. 1. Example of detected STIPs, dense sampling and STDIPs with $w = 40$ and $t = 6$ for two consecutive frames.

4. DENSE INTEREST TRAJECTORIES

Our second approach is based on the dense trajectories of Wang et al. [10]. They start by sampling feature points on a dense grid in the first frame: typically one feature every $d = 5$ pixels. Each point is tracked through the next frames by median filtering in a dense optical flow field. When a trajectory reaches a maximum length of fifteen frames, it is 'finished' and put aside for further processing. After each frame, they check whether the remaining trajectories still cover the whole image: if no tracked point is found in the neighbourhood of a grid point, a new trajectory is started there. This sampling and tracking is performed in eight scales, each one a factor $\sqrt{2}$ smaller than the previous.

We adapt this method as follows. Wang finds the starting points of his trajectories by sampling on a dense grid over a frame. We, on the contrary, look for DIPs to use as trajectory starting points. In the next step, we calculate the optical flow for all scales and we track each point in its own scale. This way, we obtain *dense interest trajectories* (DITs).

In our experiments, we use four octaves. We sample two scales per octave; each scale can be refined over four fine scales. This way, we obtain eight equivalent scales similar to Wang's method. Due to the constant size of our scales within an octave, however, we get 25% more features. The detected features likely contain more image information since they are the best-localized points in their neighbourhood. Moreover, they usually can be better tracked. These factors can be expected to give rise to a better classification performance.

A disadvantage of this strategy is the increase in processing time: we have to calculate the optical flow eight times per octave instead of the two times in [10]. Therefore, we also test without the scale selection. The location of the DIPs is then only optimized in the spatial domain.

To allow direct comparison with dense trajectories, we calculate the same four descriptors used in [10] with their code.[2] The first 28-dimensional descriptor contains the relative motion of the feature point, the others are the 96-dimensional HOG, 108-dimensional HOF and 96-dimensional MBH, all calculated locally around the trajectory.

[1] http://www.di.ens.fr/~laptev/download.html

[2] http://lear.inrialpes.fr/people/wang/dense_trajectories

Setting	YouTube		UCF Sports	
	Dense	STDIP	Dense	STDIP
$w = 10, t = 3$	73.8%	72.6%	87.9%	85.5%
$w = 10, t = 6$	74.6%	74.2%	85.0%	77.4%
$w = 10, t = 12$	72.0%	72.0%	87.3%	73.5%
$w = 20, t = 3$	70.8%	72.3%	87.3%	90.5%
$w = 20, t = 6$	70.1%	70.8%	87.4%	83.2%
$w = 20, t = 12$	67.7%	69.0%	83.5%	78.9%
$w = 40, t = 3$	65.3%	69.1%	81.4%	77.2%
$w = 40, t = 6$	61.4%	67.7%	85.3%	83.2%
$w = 40, t = 12$	59.6%	63.2%	72.0%	69.8%

Table 1. Accuracy for different spatio-temporal point methods as a function of sampling distance. The STIP accuracy is 70.1% for YouTube and 75.5% for UCF Sports.

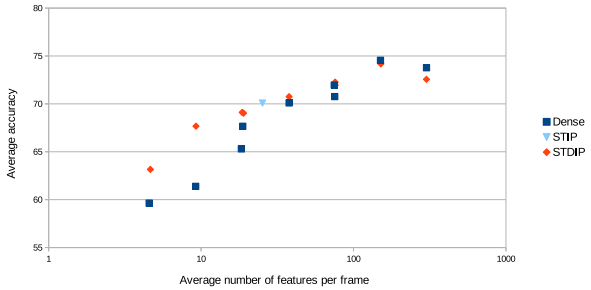

Fig. 2. YouTube: accuracy versus average number of spatio-temporal points per frame.

5. EXPERIMENTS

5.1. Datasets

We use two action recognition datasets to evaluate our features. The **YouTube dataset** [13] consists of 1600 YouTube videos of 11 action classes. Frequently, a subset of 1168 videos is selected; to be able to compare directly with [11], we use this subset as well. For evaluation, a leave-one-group-out strategy is used with 25 pre-defined groups. The final accuracy is calculated by averaging the scores of the classes.

The second dataset is the **UCF Sports Action Dataset** [14], containing 150 video samples of ten different actions. Evaluation is based on the leave-one-out strategy. The average accuracy over all action classes is the final performance measure. To increase the number of examples, we add a mirrored version of each video to the dataset. This flipped video is left out of the training data together with its original and we evaluate only on the original videos.

5.2. Action recognition setup

The following steps of the action recognition pipeline are identical for spatio-temporal dense interest points and for dense interest trajectories. First, a codebook of 4000 words is learned for each descriptor type. Next, all descriptors of one type are quantized and grouped in a bag-of-words representation. We use a multichannel support vector machine (SVM) with χ^2-kernel to combine the different descriptor types as in [15]. We train one SVM for each action class, and use one-against-all classification. We select the class with the highest probability output.

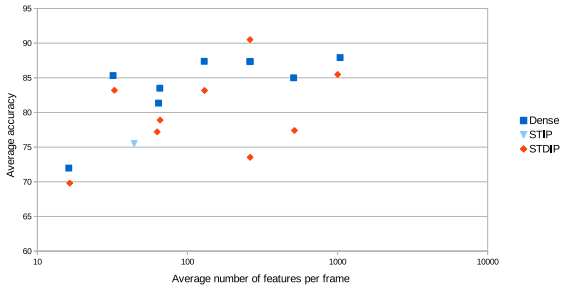

Fig. 3. UCF Sports: accuracy versus average number of spatio-temporal points per frame.

5.3. Spatio-temporal dense interest points

We test our STDIPs on the two datasets and compare them with STIP and dense sampling. Table 1 shows the classification accuracies; Figure 2 and 3 plot these accuracies against the number of features. Wang et al. [7] report on the UCF Sports dataset 78.1% for STIP and 86.1% for dense sampling. These differences can be explained by the inherent randomness of the codebook initialization.

The figure of the YouTube dataset shows that more densely sampling improves the accuracy greatly at first, however only up to a certain density. STDIPs lead to better classification than dense sampling when a similar amount of features is used, especially when sampling sparsely. When we sample more densely, STDIP and dense are very similar. Since the STDIPs have less room to be optimized and are closer to the dense points, the descriptors will be similar.

The UCF Sports results are less clear, however. Possibly, the limited number of videos makes this dataset less stable.

The number of features needed to obtain a certain accuracy can be lowered by using STDIPs. Although this is not so important in a bag-of-words setting where features can be quantized on the fly and immediately forgotten, it can be useful in cases where the location of the feature or the full feature vector has to be saved for later use (e.g., [16]).

Most of the feature detection processing time is used for the filtering step. For both STIP and STDIP, this depends on the number of scales. Since filtering is not required for dense sampling, this method is much faster. The descriptor calculation time however depends linearly on the number of features. With dense sampling, a larger number of features is needed to obtain similar classification results, so this step

d	YouTube			UCF Sports		
	Dense	DIT	DIT-NS	Dense	DIT	DIT-NS
5	83.1%	83.0%	83.1%	88.4%	89.7%	89.7%
10	83.0%	82.4%	81.9%	87.4%	90.3%	89.2%
20	79.7%	80.6%	80.9%	84.6%	89.7%	88.2%
40	73.1%	75.1%	76.9%	78.0%	78.0%	85.1%

Table 2. Accuracy for different trajectory-based methods as a function of sampling distance d.

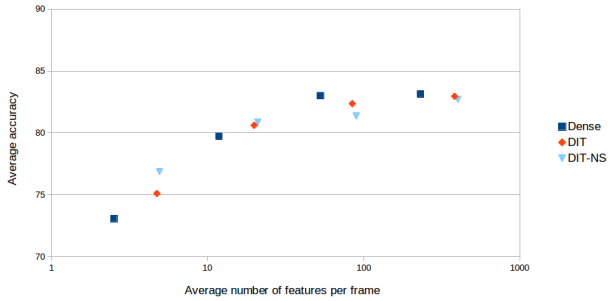

Fig. 4. YouTube: accuracy versus average number of trajectories per frame.

takes more time.

5.4. Dense interest trajectories

For the trajectory based methods, we compare dense trajectories (calculated with the code of [10]), DIT, and DIT without scale selection (DIT-NS). Each time, we experiment with four sampling densities: every 5, 10, 20 and 40 pixels. Classification results on the YouTube and UCF Sports dataset can be found in Table 2.These results are also shown in Figure 4 and 5, with on the horizontal axis the average number of features and on the vertical axis the classification accuracy. Our results are close to the 84.1% and 88.0% reported in [11] for a sampling distance of 5 pixels.

Denser sampling generally leads to more accurate classification, but this effect saturates for more than 100 features per frame. It is also clearly visible (especially for the UCF Sports data) that for equal density DITs perform better than dense sampling, and more so when less features per frame are used. If DITs are sampled more densely, they have less room to be optimized and they are more similar to the trajectories obtained by dense sampling. As a consequence, the descriptors and therefore the classification accuracy of both methods are comparable.

DIT mainly has a higher accuracy due to the larger number of features. Dense trajectories are often removed before they are finished because the tracked point is lost. Succesfully tracked points are also ignored when they move very little because they contain few information. This happens less frequently with DIT, because the start point is the easiest point to track in its environment.

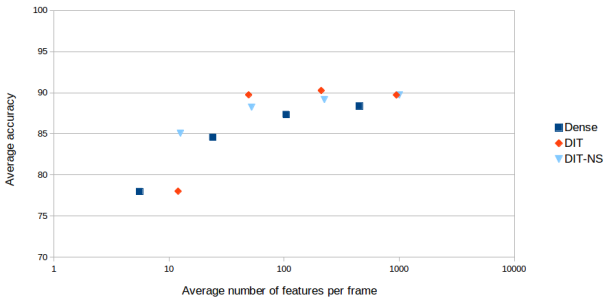

Fig. 5. UCF Sports: accuracy versus average number of trajectories per frame.

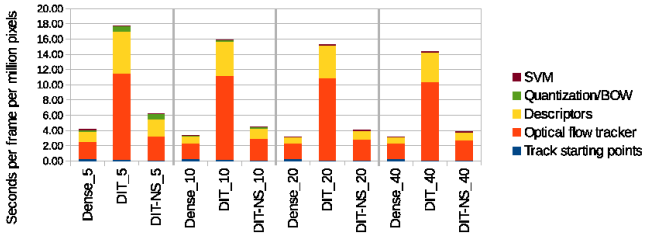

Fig. 6. Calculation time for trajectory-based methods as a function of the sample distance.

All trajectory methods score better than those based on interest points. A direct comparison is not possible, however: the used descriptors are very different. Especially the presence of the MBH descriptor could cause a large difference.

A comparison of the calculation time for the trajectory based methods for different densities is shown in Figure 6. Optical flow and descriptor calculation are most time-consuming. Since in Wang's code the optical flow is calculated over the whole image and the descriptors are based on integral images, the calculation time hardly varies when we sample less densely. With scale selection, DITs are nearly four times as slow as dense trajectories; without scale selection, however, the difference is much less, since the optical flow and integral images have to be calculated for fewer scales. Since the accuracy is almost independent of the scale selection, we recommend to not use scale selection for DIT.

6. CONCLUSION

The two presented features have advantages over interest points and dense methods. They cover the whole image and their density can easily be varied, which is not the case for interest points. On the other hand, they focus on representative and reproducable image regions, as opposed to dense sampling. In action classification, DIT and especially STDIP are better than their corresponding interest point and dense method when the sampling density is low. With high density, dense methods are more interesting since they are faster.

7. REFERENCES

[1] T. Tuytelaars, "Dense interest points," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2281–2288.

[2] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[3] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Computer Vision ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds., vol. 3954 of *Lecture Notes in Computer Science*, pp. 490–503. Springer Berlin Heidelberg, 2006.

[4] I. Laptev and T. Lindeberg, "Space-time interest points," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 432–439.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*. IEEE, 2005, pp. 65–72.

[6] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision–ECCV 2008*, pp. 650–663. Springer, 2008.

[7] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *University of Central Florida, U.S.A*, 2009.

[8] P. Matikainen, M. Hebert, and R. Sukthankar, "Trajectons: Action recognition through the motion analysis of tracked features," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 514–521.

[9] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 104–111.

[10] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[11] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[12] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[13] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1996–2003.

[14] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[15] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[16] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.